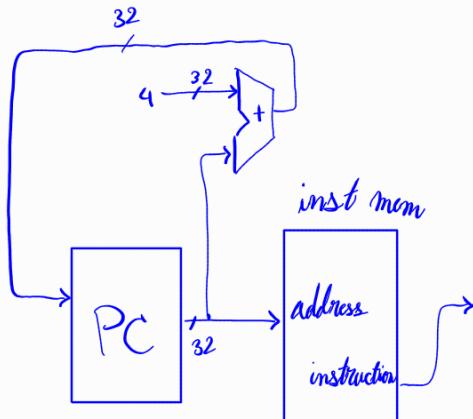


مکالمہ

”بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِيْمِ“

مکان کا میسر ہے (دعا)  
عطا کا میسر

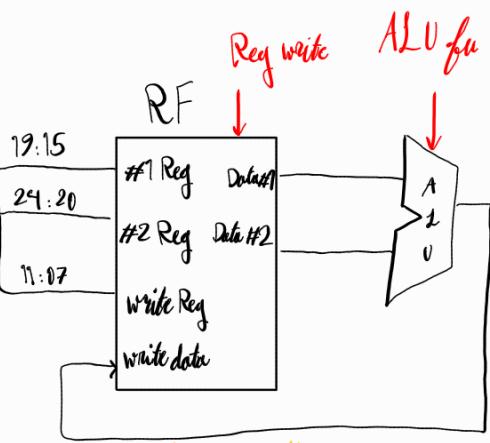
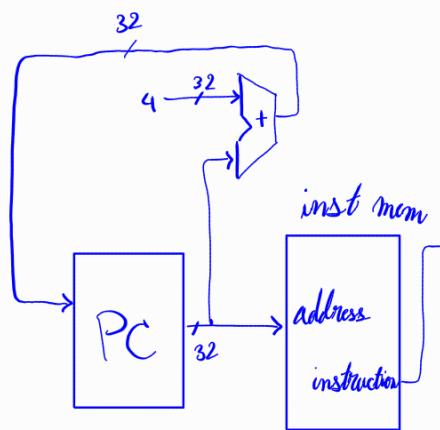
Ala Alaa Alaa



اسم مسیر داده: بخش توالی (لگ) (الستورا)

31: 25	24: 20	19: 15	14: 12	11: 7	6: 0
$f_7$	$f_t$	$r_s$	$f_3$	$f_d$	09

R-Type  $\leftarrow$   $T_{\text{new}}$  (ج) (انماذج)



odd, sub, or, slt

81:20	19:15	14:12	11:7	6:0
Imm-12b	$f_S$	$f_3$	$f_d$	op

Jahr <sup>ope</sup> 3 19 19 19 19 19 , dw, addi, xori, ori, slli I-Type (ستوراً) (اُخافِر) )

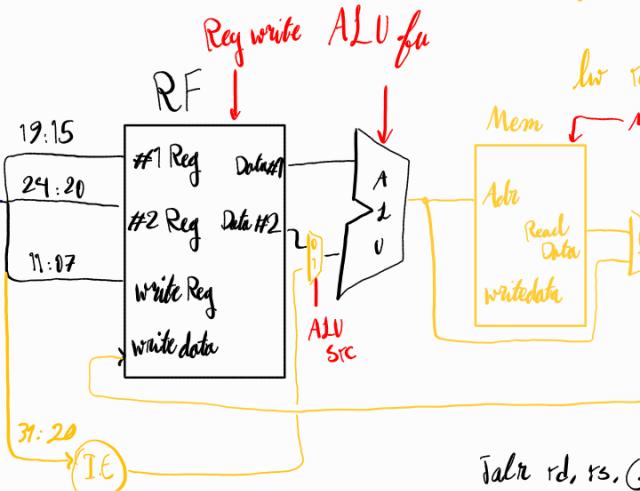
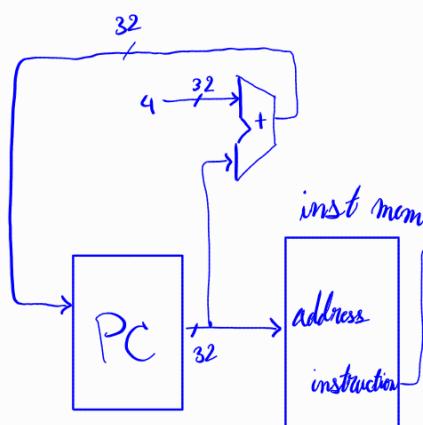


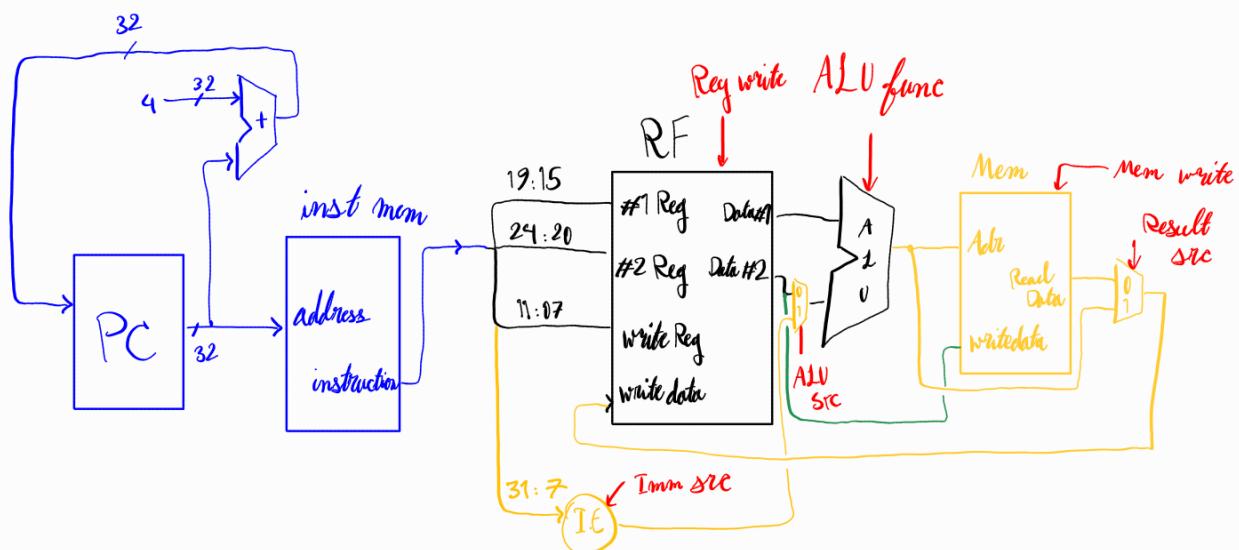
Tabelle rd, rs, (ges. 12) (ges. 1) → { rd ← pc + 4  
pc ← rs + (ges. 12) (ges. 1)

31:25 24:20 19:15 14:12 11:7 6:0

Imm[21:5] | R<sub>2</sub> | R<sub>S</sub> | f<sub>3</sub> | Imm[4:0] | OP

: S-Type (T<sub>sel</sub>) (Op)

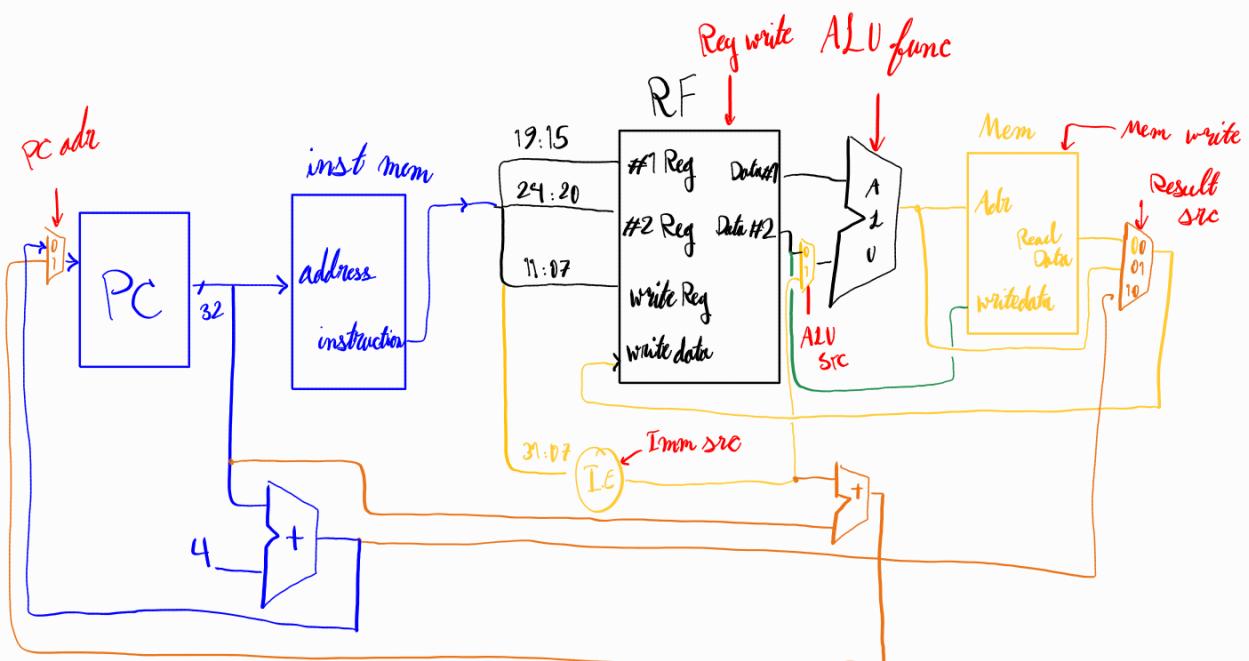
SW rd, Imm(rs)



31:12 11:7 6:0  
Imm[20], Imm[10:1], Imm[7], Imm[19:12] | Rd | OP

: J-Type (T<sub>sel</sub>) (Op)

Jal rd, adr → { rd ← PC + 4  
PC ← PC + adr



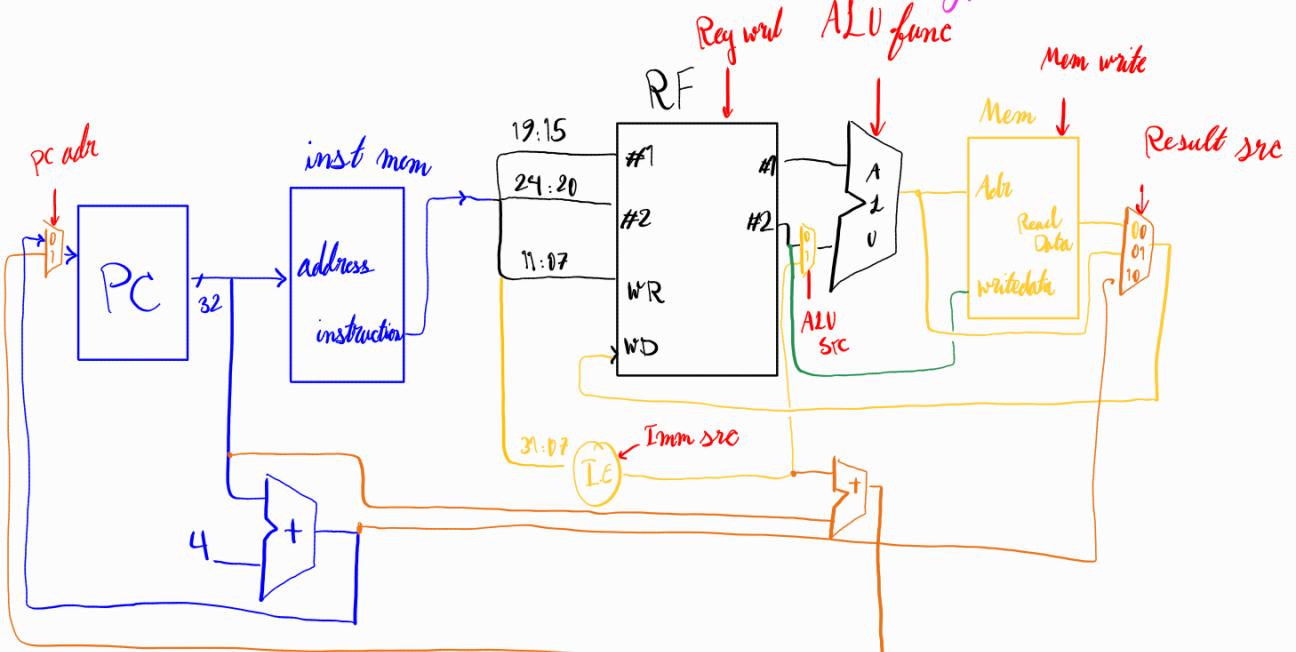
31:20

24:20 19:15 14:12 11:7 6:0

Imm[12], Imm[10:5]

r<sub>2</sub> r<sub>5</sub> f<sub>3</sub> Imm, Imm[  
14:11] [4:1]OP  
code

B-Type

T, result (Op is 10)  
bne, beq

31:12

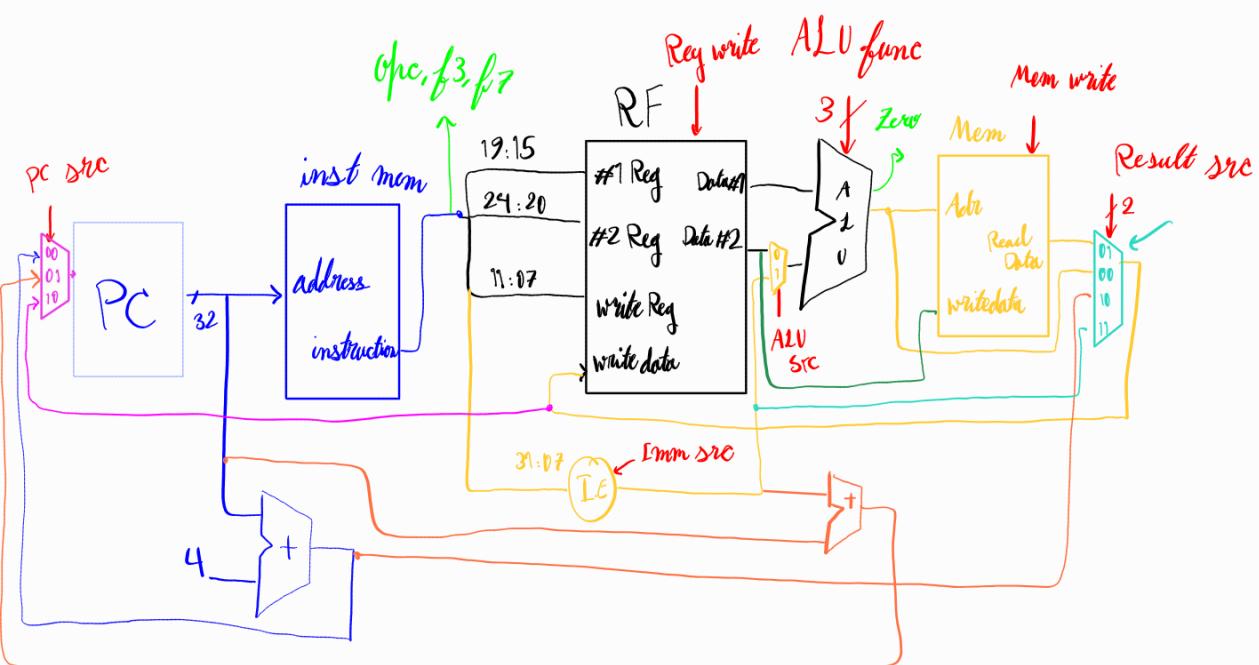
11:7 6:0

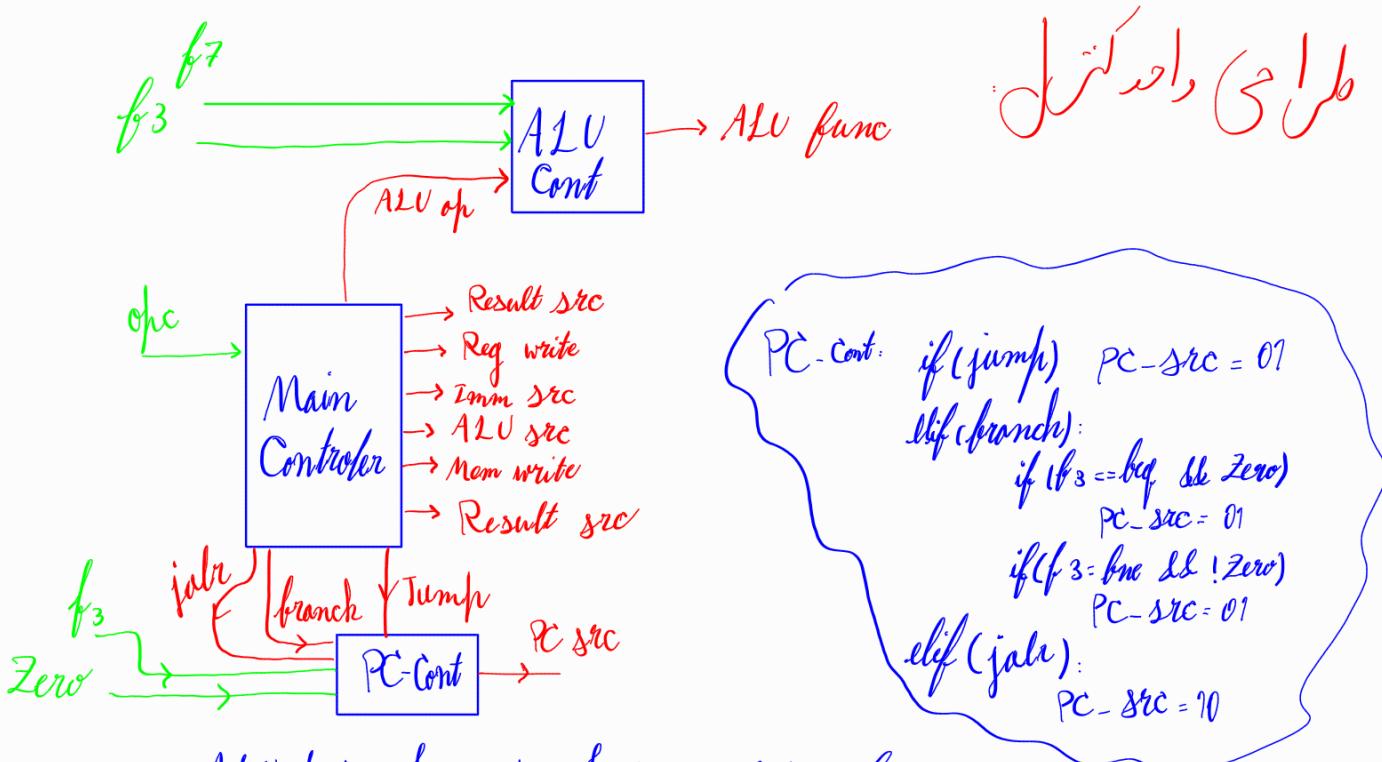
Imm[31:12]

rd

OP  
code

U-Type

result (Op is 10)  
lui rd, imm



Jalr, Branch

PC-Cont:

- if(jump) PC-Src = 01
- elif(branch):
  - if(f3 == beq & Zero) PC-Src = 01
  - if(f3 == bne & !Zero) PC-Src = 01
- elif(jalr): PC-Src = 10

ALU cont:

ALU op	f3	f7	ALU func
00	X	X	000
01	X	X	001
10		add sub and or sll xor	000 001 010 011 100 101
11		addi ori	X

Imm src:

Imm src	Type
000	{I[20:1], I[31:20]}
001	{I[20:1], I[31:25], I[11:7]}
010	{I[19:1], I[31], I[7], I[30:25], I[11:8], I[30]}
011	{I[12:1], I[19:12], I[20], I[30:21], 1'b0}
100	{I[31:12], I[12:1]}

# Main Controller

opc	Inst	Reg write	Imm src	ALU src	mem write	Result src	branch	ALU op	Jump jadr
51	R-T	1	XXX	0	0	01	0	10	0 0
3	lw	1	000	1	0	01	0	00	0 0
19	I-T	1	000	1	0	00	0	11	0 0
35	S-T	0	001	1	1	XX	0	00	0 0
111	J-T	1	011	X	0	10	0	XX	1 0
99	B-T	0	010	0	0	XX	1	01	0 0
55	U-T	1	100	X	0	11	0	XX	0 0
103	jalr	1	000	1	0	00	0	00	0 1

## \*Assembly Code:

```

int m = A[0];
for (int i=0; i<20; i++)
    if (m < A[i])
        m = A[i];
    
```

Jalr rd, rs, Goto(jwt) → { rd ← PC + 4  
PC ← rs + Goto(jwt)

```

1 main:
2     addi s0, zero, 0 # 0 is the start of array
3     lw s1, 0(s0)
4     addi s2, zero, 0
5     addi s0, s0, 4
6 loop:
7     slti t0, s2, 10
8     beq t0, zero, end_loop
9     lw t0, 0(s0)
10    slt t1, s1, t0
11    bne t1, zero, else
12    add s1, t0, zero
13 else:
14    addi s2, s2, 1
15    addi s0, s0, 4
16    jal loop
17 end_loop:
18 addi s4, zero, 0
    
```

جواب خواهد بود

